



























Projekty zaliczeniowe z laboratorium sieci komputerowych II

(semestr zimowy 2019/2020)

Michał Kalewski

20 października 2019 r.

Tematy projektów zaliczeniowych

1. Gra logiczna lub zręcznościowa, np.: szachy, warcaby, reversi itp. (1 osoba). 
2. Zdalna konsola typu *telnet* (1 osoba). 
3. System zdalnego dostępu do systemu operacyjnego Cisco IOS urządzenia sieciowego (1 osoba). 
4. System zdalnego zamykania systemów operacyjnych (1 osoba). 
5. System współdzielonego kalendarza sieciowego (1 osoba). 
6. Komunikator internetowy typu *IRC* (2 osoby). 
7. Komunikator internetowy typu *GG* (2 osoby). 
8. Komunikator internetowy typu *Skype* (2 osoby). 
9. System wymiany komunikatów typu *publish/subscribe* (2 osoby). 
10. System równoważenia obciążenia dla grupy serwerów obliczeniowych (2 osoby). 
11. System zdalnej kompresji plików (2 osoby). 
12. Grupowy edytor plików tekstowych (2 osoby). 
13. Prosty serwer protokołu HTTP zgodny ze specyfikacją RFC 2616 co najmniej w zakresie żądań: GET, HEAD, PUT, DELETE (2 osoby). 
14. Prosty serwer usługi FTP zgodny ze specyfikacją RFC 959 co najmniej w zakresie komend: *ascii*, *binary*, *mkdir*, *rmdir*, *put*, *get* (2 osoby). 
15. System poczty elektronicznej e-mail (2 osoby). 
16. System nadzoru (detekcji awarii) serwerów sieciowych z wykorzystaniem protokołów ICMP oraz TCP (2 osoby). 
17. System odkrywania i monitorowania urządzeń w sieci komputerowej (2 osoby). 
18. System robota indeksującego strony HTML wraz z aplikacją umożliwiającą wyszukiwanie stron na podstawie słów kluczowych (2 osoby). 
19. Implementacja biblioteki komunikacji niezawodnej RUDP wraz z przykładem jej wykorzystania (2 osoby). 
20. System mnożenia dużych macierzy kwadratowych (lub innych obliczeń algebry liniowej) (2 osoby). 
21. System wymiany plików w architekturze *peer-to-peer* (2 osoby). 
22. System zdalnego wywoływania procedur (2 osoby). 
23. System internetowego nadzoru wizyjnego (2 osoby). 
24. Implementacja wybranego (lub autorskiego) protokołu routingu dynamicznego (2 osoby). 
25. Zdalnie konfigurowana zaporę sieciową (2 osoby). 
26. Implementacja dowolnej (lecz uzgodnionej) usługi sieciowej zgodnie ze specyfikacjami dokumentów RFC¹. 

Uwagi

- Wszystkie projekty należy zrealizować w **architekturze klient-serwer** (wyjątkiem jest zadanie nr 21) z użyciem **protokołu TCP** (wyjątkiem jest zadanie nr 19).
- Implementacje serwerów **współbieżnych** należy wykonać dla systemów operacyjnych GNU/Linux z użyciem języka C lub C++, a implementacje aplikacji klientów dla systemów Microsoft Windows (nie wolno stosować systemu Cygwin)

¹<http://www.ietf.org/rfc.html>

z użyciem języka C, C++, C#, Java lub Python (zastosowanie innego języka programowania wymaga uzyskania zgody) oraz z graficznym interfejsem użytkownika.

- Dopuszcza się także możliwość implementacji aplikacji klientów dla urządzeń mobilnych z systemami operacyjnymi Android oraz iOS.
- Dodatkowe informacje na temat projektów można uzyskać podczas każdego zajęcia laboratoryjnych oraz w czasie konsultacji.
- Należy przestrzegać poniższych zasad zaliczania projektów!

Podstawowe zasady zaliczania projektów

- Kody projektów muszą być utrzymywane w repozytorium git² w systemie GitLab: <https://gitlab.cs.put.poznan.pl>.
- Wszystkie programy muszą się poprawnie i bez ostrzeżeń (z opcją `-Wall` dla języków C i C++) kompilować na komputerach laboratoryjnych.
- Podczas zaliczania, programy będą uruchamiane tylko na komputerach laboratoryjnych (wyjątkiem są urządzenia mobilne).
- Programy muszą być napisane w sposób czytelny i przejrzysty, należy umieszczać stosowne komentarze w kodzie, a także warto stosować styl kodowania wypracowany przez firmę Google³.
Uwaga: programy napisane w sposób nieczytelny nie będą zaliczane!
- Do każdego projektu należy dołączyć krótkie sprawozdanie (maksymalnie dwie strony formatu A4) w formacie PDF, które ma zostać umieszczone w repozytorium projektu zaliczeniowego. W sprawozdaniu tym należy zawrzeć następujące informacje:
 - temat zadania,
 - **opis protokołu komunikacyjnego**,
 - opis implementacji, w tym krótki opis zawartości plików źródłowych,
 - sposób kompilacji, uruchomienia i obsługi programów projektu.
- **Co najmniej dwa dni przed planowanym terminem zaliczania projektu należy wysłać e-mail do prowadzącego zajęcia zawierający nazwę oraz adres repozytorium projektu zaliczeniowego w systemie GitLab.** W temacie wiadomości e-mail proszę wpisać: [SK2] Projekt zaliczeniowy.
Uwaga: należy dodać prowadzącego zajęcia jak członka grupy projektu w systemie GitLab z uprawnieniami „Developer”.
- **Ostatecznym terminem zaliczania projektów są przedostatnie zajęcia laboratoryjne!**

Kryteria oceny projektów

- Poprawność implementacji projektu, a w szczególności komunikacji sieciowej.
- Poprawność protokołu komunikacyjnego.
- Zgodność funkcjonalności projektu z uzgodnionymi wymaganiami.
- Przejrzystość i czytelność kodu.
- Inicjatywa i pomysłowość w realizacji projektu.
- Przestrzeganie zasad zaliczania projektu.

²git Documentation: <http://git-scm.com/doc>

³Google C++ Style Guide: <https://google.github.io/styleguide/cppguide.html>